ロボット工学第二２００４　補遺
倉林大輔

3　目標軌道

「制御」という言葉で国語辞典を引くと、「押さえつけて思い通りにすること」とある。目標軌道を作るということは、目標とする「思い通り」の方を作るということになる。一般に制御は目標に対する誤差をなくそうとする問題を扱っているが、その目標をどのように生成するかというのは、これでまた奥の深い問題である。このような問題は「計画問題」といわれて「制御問題」と区別されることも多い。

本章では、まず既に通るべき幾何学的な経路（曲線）が与えられているとして、その上での時間的な計画について紹介する。ついで、障害物等を避けて初期状態から目標状態へ至る為の幾何学的軌道計画法について紹介する。

3.1　関節角目標軌道

目標軌道とは、制御対象となるマニュピレータの各自由度について、位置・速度・加速度の時間的な履歴を決定しておくものである。目標軌道の与え方としては、関節角空間で直接各関節の目標となる動きを決めるものと、手先空間で作業から直接目標軌道を与えるものの２種がある。関節角空間で目標軌道を与えると、マニュピレータの手先がどのような動きをするか直接知ることができない。その一方で、特異点などは問題とならない。一方、手先空間で目標軌道を与える場合、手先の挙動を直接指定できるのが大きなメリットである。しかし、特異点の回避や、到達可能性について事前に調べておくことが必要となる。

いま、時間を t とおき、関節角からなるベクトルを $\theta(t)$ としよう。初期条件・終端条件を (1) とおくと、$\theta(t)$ が３次関数 (2) であれば、条件を満たす軌道を表現可能である。

$$\begin{cases} \theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \\ \dot{\theta}(0) &= \dot{\theta}(t_f) = 0 \end{cases} \tag{1}$$

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{2}$$

条件を代入して、３次関数の係数を決定すると (3) となる。なお、加速度を条件として指定するなら５次関数を用意すれば良い。

$$\begin{cases} a_0 &= \theta_0 & a_1 &= 0 \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) & a_3 &= -\frac{2}{t_f^3}(\theta_f - \theta_0) \end{cases} \tag{3}$$

3.2　手先空間での目標軌道

手先空間での目標軌道は、関節空間でのそれと本質的に変わることはないが、出発点と終端点の間を補間する形で目標軌道を生成する場合は以下の点に注意が必要である。

1. 軌道途中で到達不可能点に至らないか：出発点と終端点が到達可能領域内にあっても、その途中が到達可能であるとは限らない（図 1(a)）。
2. 特異点を通らないか：特異点では制御入力が無限大に発散してしまうので、事実上通過することができない（図 1(b)）。
3. 初期姿勢と終端姿勢が別の解である。すなわち、目標軌道は連続であるが、マニピュレータが連続的に目標軌道を追従することができず、途中で１度大きく姿勢を変える必要がある場合（図 1(c)）、事実上目標軌道を達成できない。

正しく目標軌道が与えられ、特異点を通過していなければ、次章以降で示すように手先での目標軌道と関節角での目標軌道に、制御上での本質的な差はない。

図 1: 実現が不可能か、困難であるような目標軌道

3.3 手先姿勢に対する目標姿勢

手先の姿勢について、時間に対する変化状態を指定したい場合がある。このとき、連続に手先の姿勢を変化させるよう関数で表現するのは意外と難しい。ここでは２種の表現方法を紹介する。

3.3.1 １軸回転法

原点を共有する（平行移動のない）２つの座標系 Σ_A, Σ_B の間の回転変換は、いままで複数の軸の回転変換の合成で表現してきたが、実際にはある空間上の軸まわりに一定量回転させることと等価である。この回転軸を**等価回転軸**とよび、$k = [k_x,\ k_y,\ k_z]$ と書くことにしよう。この軸回りに α 回転させることで２つの座標系が一致するならば、等価回転軸と回転量の対 $[k,\ \alpha]$ も回転変換を表している。実はこの表現は冗長（２つの座標系間の回転変換を表すには３変数あれば十分）であるので、等価回転軸と同じ向きを持ち、長さが α であるようなベクトルを用いてこの回転を表現する場合が多い。

ある単位長さベクトル k まわりに α だけ回転させる変換行列を $R(k,\ \alpha)$ と書くことにすると、この行列は (4) と書き表される。

$$R(k,\alpha) = \begin{bmatrix} k_x^2 V_\alpha + C_\alpha & k_x k_y V_\alpha - k_z S_\alpha & k_x k_z V_\alpha + k_y S_\alpha \\ k_x k_y V_\alpha + k_z S_\alpha & k_y^2 V_\alpha + C_\alpha & k_y k_z V_\alpha - k_x S_\alpha \\ k_x k_z V_\alpha - k_y S_\alpha & k_y k_z V_\alpha + k_x S_\alpha & k_z^2 V_\alpha + C_\alpha \end{bmatrix} \tag{4}$$

ただし、V_α は $1 - C_\alpha$ を表す。

この表現を用いれば、時間の関数となる変数は α のみであるから、α に関して適切な境界条件を設定し、軌道を決定すれば良い。

3.3.2 ２軸回転法

１軸回転法は数学的には冗長性がなく完成された表現であるが、２つの手先姿勢の間でどのような回転が起こるのか、人間にとって直感的に理解しずらいという問題があった。これに対し、ZYZ オイラー角表現を利用して２軸の周りに回転させる方法が考えられる。これが２軸回転法である。

一般にマニュピレータでは、手先の作用方向（人間でいえば指先方向）に Z 軸が取られるため、この方向では Z 軸回りと Y 軸回りの回転を合成する。人間にとってみると、手のひらの回転と手首の回転が独立しているように見え、直感的に理解しやすいと言われている。

いま、手先の初期姿勢（を表す座標系）と終端姿勢の間が ZYZ オイラー角 $(\phi,\, \theta,\, \psi)$ で表されているとする。このとき、このオイラー角表現は (5) のように分割することができる。

$$R(\hat{k},\, \theta)R(\hat{Z}, \phi + \psi) \tag{5}$$

ここで、$\hat{k} = [-S_\phi, C_\phi, 0]^T$、また $\hat{Z}$ は $R(\hat{k},\, \theta)$ で変換した後の Z 軸を表す。

よって、$\theta = \alpha(t),\ \phi + \psi = \beta(t)$ として、(6) として変換を時間の関数とすることができる。

$$R(\hat{k},\, \alpha(t))R(\hat{Z}(t), \beta(t)) \tag{6}$$

ただし、$\hat{Z}(t)$ は $R(\hat{k},\, \alpha(t))$ で変換された Z 軸を表す。$\alpha(t),\ \beta(t)$ はオイラー角表現において適切な終端条件を与えて求めればよい。この表現は式の上では面倒に見えるが、固定軸 $\hat{k}$ まわりに回転を行いつつ、そのときの手先方向に対して手先を回転させればよいので、手先の向きを理解しやすい（といわれている）。

3.4 コンフィグレーション空間

ロボットの姿勢（あるいは手先の位置と向き）を初期状態から目標状態へ変化させるとき、そのマニュピレータの周囲に障害物等が全く無ければ、目標位置に向かって静定制御を行えば十分である（制御については次節で述べる）。しかし現実には、様々な障害物が存在することが普通である。たとえば、自動車工場において車体へダッシュボードを取り付けるロボットの場合、複雑な形をした車体内部へアームを衝突させないようにしながら進入させなくてはならない。このような場合、障害物を回避するための目標軌道をあらかじめ作っておくことが必要になる。

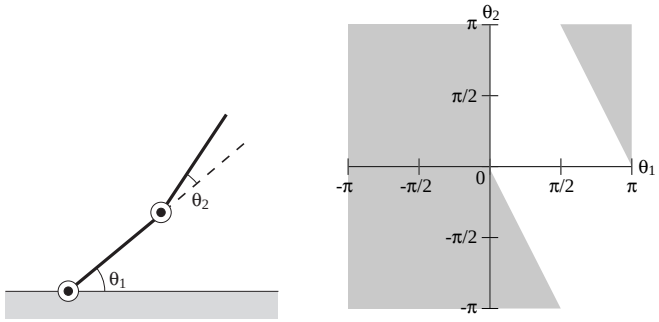
このような場合、一般化座標を用いて周囲の空間を表現してしまえば、その表現における自由空間と実空間における自由空間が一対一に対応し便利である。このような空間表現を**コンフィグレーション空間**（**配位空間**との訳語もあるようだ）と呼ぶ。

この空間では、マニピュレータがある姿勢をとる、つまり一般化座標における点を占めた場合に、マニピュレータが障害物と衝突しないか、あるいはそもそもその姿勢をとることができないか、などを判定していく。そして、障害物あるいは取ることのできない姿勢であった場合、コンフィグレーション空間内のその点は、「障害物内部」として扱われる。このようにすることで、実空間での障害物をコンフィグレーション空間に投影することができる。これらコンフィグレーション空間内での障害物表現を**コンフィグレーション障害物**（C障害物）と呼ぶ。

コンフィグレーション空間表現は、一旦作ってしまえば大変便利だが、マニピュレータの自由度が上昇すると、空間の次元も上昇し、コンフィグレーション障害物算出の演算コストが爆発的に増大することが問題である。

3.5 幾何的目標軌道

ロボットを障害物の存在する空間で動かそうとした場合、障害物を避けつつ目標に向かう空間的な軌道を決める必要がある。幾何的目標軌道生成（計画）法はロードマップ法・セル分割法・ポテンシャル法の３種に大別される。ここでは代表的な計画法を紹介する。



(a) 2 リンクマニピュレータ　　(b) 地面のC障害物

図 2: コンフィグレーション空間における障害物の例：長さが等しい (a) の 2 リンクマニピュレータにおいて、「地面」をコンフィグレーション空間に投影すると (b) になる。

3.5.1 ロードマップ法

与えられた空間（障害物含む）に、通過可能な「軌道の候補」を複数作り、それらが形成するグラフ構造の上で経路探索を行う方法である。代表的なものに可視グラフ法やボロノイ線図法がある。可視グラフ法は、状態空間が 2 次元平面で表現可能な場合は最短の経路を与える。空間が 3 次元以上の場合はその保証が無いので注意が必要である。ボロノイ線図法は、障害物から最も距離が遠い点の集合を利用するので、最も「安全な」経路を生成する。

ロードマップ法の問題点は、多次元空間になると計算量が爆発的に増大したり、2 次元のとき通用した最適化が使えなくなるといったことである。

3.5.2 セル分割法

セル分割法とは、与えられた空間を何らかの基準でより細かな・かつ性質の明らかな小領域（「セル」という）に分割し、それらの連なりによって障害物などと衝突しない幾何学的経路を算出するものである。ここでいう「性質」とは、（A）完全な自由空間、（B）完全に障害物と干渉する空間、（C）(A) でも (B) でもない空間、の 3 種である。初期状態から目標状態まで（A）のセルの連なりを見つけることができれば、あとは一定の規則に従ってそのセルを通過する軌道を作っていけばよい。

セル分割法の問題点はアルゴリズムの完全性が保障できないことである。

3.5.3 ポテンシャル法

ポテンシャル法は、与えられた空間に重力や電磁力に見立てたポテンシャル場を一定の関数により生成し、ポテンシャルの勾配に従って制御点を目標まで動作させる手法である。いわば、空間にピンボール台のように勾配をつけて、ボールを目標状態まで導くようなものである。ポテンシャルの生成にはいくつか方法があり、ラプラスポテンシャルを生成すれば停留点なく目標状態へ到達できるが、これは非常に計算量が多い。これと比べて簡単なポテンシャル生成法は数多く提案されているが、いずれも停留点の存在を排除しきれないことが弱点といえる。